

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-94567

(P2002-94567A)

(43)公開日 平成14年3月29日 (2002.3.29)

(51)Int.Cl.⁷

H 04 L 12/56

識別記号

2 0 0

F I

H 04 L 12/56

コード*(参考)

2 0 0 Z 5 K 0 3 0

審査請求 未請求 請求項の数38 OL (全 21 頁)

(21)出願番号 特願2001-242612(P2001-242612)

(22)出願日 平成13年8月9日(2001.8.9)

(31)優先権主張番号 09/635988

(32)優先日 平成12年8月9日(2000.8.9)

(33)優先権主張国 米国(US)

(71)出願人 391055933

マイクロソフト コーポレイション
MICROSOFT CORPORATION

アメリカ合衆国 ワシントン州 98052-
6399 レッドモンド ワン マイクロソフ
ト ウェイ (番地なし)

(72)発明者 トマス ディー. ブラウン

アメリカ合衆国 94065 カリフォルニア
州 レッドウッド ショアーズ ブリッジ
パークウェイ 3600 スイート 200

(74)代理人 100077481

弁理士 谷 義一 (外2名)

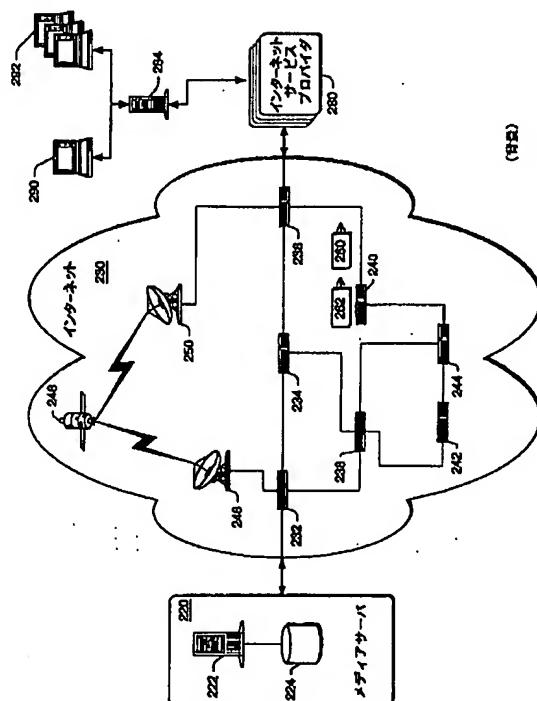
最終頁に続く

(54)【発明の名称】 TCPネットワーク環境におけるバンド幅の高速動的測定

(57)【要約】

【課題】 パケットペア手法を使用することの可能なTCPネットワーク環境におけるバンド幅の高速動的測定方法を提供する。

【解決手段】 バンド幅の高速動的測定では、これらのアルゴリズムで調されている遅延を解消するための対応策が取り入れられている。その対応策とは、Nagleアルゴリズムの適用を禁止することと、パケットペアの直後に「プッシュ」パケットを送信することによってパケットのバッファリングを最小限にすること、およびダミーパケットでプライミング(初期化)することによってSlow Startアルゴリズムを回避することである。



【特許請求の範囲】

【請求項1】 ネットワーク上のエンティティ間の高速パケット通信を容易化する方法であって、

ディレイディスエーブル (delay-disable) コマンドを送信し、

パケット集合を送信側エンティティから受信側エンティティに送信することを特徴とする方法。

【請求項2】 請求項1に記載の方法において、パケット集合はバックツーバックで送信される2つのパケットを含むことを特徴とする方法。

【請求項3】 請求項1に記載の方法において、パケット集合は、第1バンド幅測定パケットと第2バンド幅測定パケットとを含み、第2パケットは第1パケットの直後に送信されることを特徴とする方法。

【請求項4】 請求項1に記載の方法において、ネットワークはTCPであることを特徴とする方法。

【請求項5】 請求項1に記載の方法において、ディレイディスエーブルコマンドは、ネットワーク上の1または2以上の通信デバイス上のNagleアルゴリズムを使用禁止することを特徴とする方法。

【請求項6】 請求項1に記載の方法において、ディレイディスエーブルコマンドはTCP_NODELAYであることを特徴とする方法。

【請求項7】 コンピュータによって実行されたとき、請求項1に記載の方法をOSIモデルに準拠するアプリケーション層で実行するコンピュータ実行可能命令を実装していることを特徴とするプログラムモジュール。

【請求項8】 コンピュータによって実行されたとき、請求項1に記載の方法を実行するコンピュータ実行可能命令を収めていることを特徴とするコンピュータ読取可能媒体。

【請求項9】 ネットワーク上のエンティティ間の高速パケット通信を容易化する方法であって、

パケット集合を送信側エンティティから受信側エンティティに送信し、この場合、パケット集合に含まれるパケット間の伝送遅延は許容限度を超えており、

前記の直後に、パケット集合に含まれるパケット間の伝送遅延を警告するために少なくとも1つの「プッシュ」パケットを送信し、この場合、前記遅延はネットワーク上の通信デバイスのパケットバッファリングが原因であることを特徴とする方法。

【請求項10】 請求項9に記載の方法において、パケット集合は、バックツーバックで送信される2つのパケットを含むことを特徴とする方法。

【請求項11】 請求項9に記載の方法において、パケット集合は送信側エンティティと受信側エンティティ間のバンド幅を測定するためのバンド幅測定パケットであることを特徴とする方法。

【請求項12】 請求項9に記載の方法において、通信デバイスはプロキシサーバであることを特徴とする方

法。

【請求項13】 請求項9に記載の方法において、ネットワークはTCPであることを特徴とする方法。

【請求項14】 コンピュータによって実行されたとき、請求項9に記載の方法をOSIモデルに準拠するアプリケーション層で実行するコンピュータ実行可能命令を実装していることを特徴とするプログラムモジュール。

【請求項15】 コンピュータによって実行されたとき、請求項9に記載の方法を実行するコンピュータ実行可能命令を収めていることを特徴とするコンピュータ読取可能媒体。

【請求項16】 ネットワーク上のエンティティ間の高速パケット通信を容易化する方法であって、
パケット集合を送信側エンティティから受信側エンティティに送信し、この場合、パケット集合に含まれるパケット間の伝送遅延は許容限度を超えており、
前記の直後に、パケット集合に含まれるパケット間の伝送遅延を警告するために少なくとも1つの「プライミング」パケットを送信し、この場合、前記遅延はネットワーク上の通信デバイスのフロー制御機能が原因であることを特徴とする方法。

【請求項17】 請求項16に記載の方法において、パケット集合は、バックツーバックで送信される2つのパケットを含むことを特徴とする方法。

【請求項18】 請求項16に記載の方法において、パケット集合は送信側エンティティと受信側エンティティ間のバンド幅を測定するためのバンド幅測定パケットであることを特徴とする方法。

【請求項19】 請求項16に記載の方法において、ネットワークはTCPであることを特徴とする方法。

【請求項20】 請求項16に記載の方法において、送信側エンティティから受信側エンティティへの間にTCPコネクションを確立するステップをさらに含み、確立するステップはパケット集合が送信される直前に行われることを特徴とする方法。

【請求項21】 請求項16に記載の方法において、フロー制御機能はSlow Startアルゴリズムであることを特徴とする方法。

【請求項22】 コンピュータによって実行されたとき、請求項16に記載の方法をOSIモデルに準拠するアプリケーション層で実行するコンピュータ実行可能命令を実装していることを特徴とするプログラムモジュール。

【請求項23】 コンピュータによって実行されたとき、請求項16に記載の方法を実行するコンピュータ実行可能命令を収めていることを特徴とするコンピュータ読取可能媒体。

【請求項24】 ネットワーク上の2エンティティ間のバンド幅測定を容易化する方法であって、

ディレイディスエーブル (delay-disable) コマンドを送信し、

送信側エンティティから受信側エンティティにペアのバンド幅測定パケットを送信することを特徴とする方法。

【請求項25】 請求項24に記載の方法において、ペアのパケットに関係する測定に基づくバンド幅計算を受信するステップをさらに含むことを特徴とする方法。

【請求項26】 ネットワーク上のエンティティ間のバンド幅測定を容易化する方法であって、

ペアのパケットを送信側エンティティから受信側エンティティに送信し、この場合、ペアに含まれるパケット間の伝送遅延は許容限度を超えており、

前記の直後に、ペアに含まれるパケット間の伝送遅延を警告するために少なくとも1つの「プッシュ」パケットを送信し、この場合、前記遅延はネットワーク上の通信デバイスのパケットバッファリングが原因であることを特徴とする方法。

【請求項27】 請求項26に記載の方法において、ペアのパケットに関係する測定に基づくバンド幅計算を受信することを特徴とする方法。

【請求項28】 ネットワーク上のエンティティ間のバンド幅測定を容易化する方法であって、

ペアのパケットを送信側エンティティから受信側エンティティに送信し、この場合、ペアに含まれるパケット間の伝送遅延は許容限度を超えており、

前記の直後に、ペアに含まれるパケット間の伝送遅延を警告するために少なくとも1つの「プライミング」パケットを送信し、この場合、前記遅延はネットワーク上の通信デバイスのフロー制御機能が原因であることを特徴とする方法。

【請求項29】 請求項28に記載の方法において、ペアのパケットに関係する測定に基づくバンド幅計算を受信するステップをさらに含むことを特徴とする方法。

【請求項30】 コンピュータによって実行されたとき、ネットワーク上のエンティティ間の高速パケット通信を高速化する方法を実行するコンピュータ実行可能命令を収めているコンピュータ読取可能媒体であって、前記方法は、

ディレイディスエーブル (delay-disable) コマンドを送信し、

送信側エンティティから受信側エンティティにパケット集合を送信することを特徴とするコンピュータ読取可能媒体。

【請求項31】 コンピュータによって実行されたとき、ネットワーク上のエンティティ間の高速パケット通信を高速化する方法を実行するコンピュータ実行可能命令を収めているコンピュータ読取可能媒体であって、前記方法は、

パケット集合を送信側エンティティから受信側エンティティに送信し、この場合、パケット集合に含まれるパケ

ット間の伝送遅延は許容限度を超えており、

前記の直後に、パケット集合に含まれるパケット間の伝送遅延を警告するために少なくとも1つの「プッシュ」パケットを送信し、この場合、前記遅延はネットワーク上の通信デバイスのパケットバッファリングが原因であることを特徴とするコンピュータ読取可能媒体。

【請求項32】 コンピュータによって実行されたとき、ネットワーク上のエンティティ間の高速パケット通信を高速化する方法を実行するコンピュータ実行可能命令を収めているコンピュータ読取可能媒体であって、前記方法は、パケット集合を送信側エンティティから受信側エンティティに送信し、この場合、パケット集合に含まれるパケット間の伝送遅延は許容限度を超えており、前記の直後に、パケット集合に含まれるパケット間の伝送遅延を警告するために少なくとも1つの「プライミング」パケットを送信し、この場合、前記遅延はネットワーク上の通信デバイスのフロー制御機能が原因であることを特徴とするコンピュータ読取可能媒体。

【請求項33】 プロセッサと、プロセッサ上で実行可能な伝送遅延アボイダとを備えた装置であって、前記伝送遅延アボイダは、

ディレイディスエーブル (delay-disable) コマンドを送信し、
送信側エンティティから受信側エンティティにパケット集合を送信することを特徴とする装置。

【請求項34】 プロセッサと、プロセッサ上で実行可能な伝送遅延アボイダとを備えた装置であって、前記伝送遅延アボイダは、

パケット集合を送信側エンティティから受信側エンティティに送信し、この場合、パケット集合に含まれるパケット間の伝送遅延は許容限度を超えており、前記の直後に、パケット集合に含まれるパケット間の伝送遅延を警告するために少なくとも1つの「プッシュ」パケットを送信し、この場合、前記遅延はネットワーク上の通信デバイスのパケットバッファリングが原因であることを特徴とする装置。

【請求項35】 プロセッサと、プロセッサ上で実行可能な伝送遅延アボイダとを備えた装置であって、前記伝送遅延アボイダは、

パケット集合を送信側エンティティから受信側エンティティに送信し、この場合、パケット集合に含まれるパケット間の伝送遅延は許容限度を超えており、前記の直後に、パケット集合に含まれるパケット間の伝送遅延を警告するために少なくとも1つの「プライミング」パケットを送信し、この場合、前記遅延はネットワーク上の通信デバイスのフロー制御機能が原因であることを特徴とする装置。

【請求項36】 データフィールドがそこに符号化されていて、通信チャネル上を伝送される変調データ信号であって、

ディレイディスエーブル (delay-disable) コマンドを含んでいる第1フィールドと、第1バンド幅測定パケットを含んでいる第2フィールドと、

第2バンド幅パケットを含んでいる第3フィールドと、を備えていることを特徴とする変調データ信号。

【請求項37】 データフィールドがそこに符号化されていて、通信チャネル上を伝送される変調データ信号であって、

第1バンド幅測定パケットを含んでいる第1フィールドと、

第1バンド幅測定パケットを含んでいる第2フィールドと、

第1パケットと第2パケット間の伝送遅延の最小限化を容易にする「アッシュ」パケットを含んでいる第3フィールドとを備え、前記遅延はネットワーク上の通信デバイスのパケットバッファリングが原因であることを特徴とする変調データ信号。

【請求項38】 データフィールドがそこに符号化されていて、通信チャネル上を伝送される変調データ信号であって、

「プライミング」パケットを含んでいる第1フィールドと、

第1バンド幅測定パケットを含んでいる第2フィールドと、

第2バンド幅測定パケットを含んでいる第3フィールドとを備え、前記「プライミング」パケットはパケット間の伝送遅延の最小限化を容易にし、前記遅延はネットワーク上の通信デバイスのフロー制御機能が原因であることを特徴とする変調データ信号。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、TCPネットワーク環境上のエンティティ間のコネクションのための最大バンド幅を動的に検出することに関する。具体的には、本発明は、あるパケット集合の伝送を効果的に遅延させる可能性のある、ネットワーク環境のフロー制御機能に対する対応策に関する。

【0002】

【従来の技術】 インターネット (the Internet) の成熟に伴って、インターネット上で利用可能なコンテキストの特性が変化している。現在では、サウンドとビデオコンテンツが従来のテキストコンテンツに組み込まれている。しかし、インターネット上のこの新コンテンツでは、数年前までは普通に利用可能であったものよりも、コネクション速度 (つまり、バンド幅) の高速化が要求されている。

【0003】 図1は、代表的なインターネットの構成の例を示す。この構成には、サーバ (メディアサーバ20など) が含まれ、このサーバはインターネット30に結

合されている。サーバには、1つまたは2つ以上の物理的サーバコンピュータ22が置かれているのが代表的であり、サーバコンピュータは1つまたは2つ以上の物理的ストレージデバイスおよび/またはデータベース24を装備している。インターネット伝送の反対側には、クライント90、92が存在し、これは多数の利用可能なインターネットサービスプロバイダ (Internet Service Provider (ISP))

80の1つを通して接続されている。ここでは、サーバはデータを送信するネットワークエンティティであり、クライアントはデータを受信するネットワークエンティティになっている。

【0004】 クラウド (集團、cloud) 30はインターネットと明記されているが、理解されているように、このクラウドは、図示のものだけを含んでいるインターネット部分を表している。このクラウドの内側には、ルータ、伝送回線、コネクション、および他の通信デバイスが置かれていて、クライアントとサーバの間でデータがほとんど正常に伝送されるようにしている。例示のインターネットクラウド30の内側には、ルータ32～44、2つのサテライトディッシュ (satellite dish) 46、50およびサテライト48が存在している。これらのデバイス間のリンクは取り得る経路 (path) を表し、データパケットは、その経路を通じてサーバとクライアント間を伝送されるようになっている。

【0005】 一般的に、ネットワーク (インターネットなど) 上の通信デバイスは、ネットワークを利用した2エンティティ間のコミュニケーションを容易にするデバイスになっており、そこには、2つのエンティティが含まれている。そのようなエンティティの例としては、サーバ20とクライント90がある。

【0006】 OSIモデルの層

開放型システム間相互接続 (Open System Interconnection (OSI)) モデルは、世界的規模のコミュニケーションのためのISO標準であり、そこでは、プロトコルを7層で実装するためのネットワーキングフレームワークが定義されている。コントロールは、あるステーション (局) に置かれたアプリケーション層から始まって、ある層から次の層へ渡され、最下層に移った後、チャネルを通じて次のステーションに渡され、階層の上方に向かって戻るようになっている。当業者ならば、このOSIモデルは周知である。

【0007】 OSIモデルに含まれる機能の大部分は、2または3つのOSI層が1つに統合されていることがあっても、すべての通信システムに存在している。これらの層は「レベル」とも呼ばれている。

【0008】 一般的に、物理層はハードウェアで実現されている。そのようなハードウェアとしては、ネットワ

ークカード、モ뎀、またはある種の他の通信デバイスがある。代表例として、トランスポート層は、オペレーティングシステム (operating system OS) のカーネルで実現されている。

【0009】【0009】スタックの最上層はアプリケーション層であり、そこにはアプリケーションが置かれている。ここには、Webブラウザやメディアプレイヤ、eメールプログラムのように、コンピュータの外側に置かれたエンティティと通信するアプリケーションが置かれている。アプリケーション層では、インターネットのようなネットワーク上のエンティティ間のコミュニケーション詳細に対する制御権は、最小になっている。

【0010】バンド幅

バンド幅 (bandwidth) とは、一定時間内に伝送できるデータ量のことである。例えば、図1に示すメディアサーバ20とメディアクライアント90間のバンド幅は、ある時間単位（例えば、1秒）内に両者間で伝送できるデータ量（例えば、1000ビット）で計算される。もっと具体的に説明すると、データは、概算で秒当たり56,000ビットのレートでデバイス間を伝送することができる。このレートは、毎秒56キロビット (Kbps) で表すこともできる。

【0011】図1に示すように、インターネット上の伝送は複数のリンクを通り抜けてから、そのデスティネーション（目的地）に到達する。各リンクは独自のバンド幅をもっている。チェーン（鎖）はその最も弱いリンクと同じ強度しかないのと同じように、サーバ20とクライアント20間の最大バンド幅は、最低速バンド幅をもつリンクで両者間を結んでいくことになる。代表的には、これはクライアント90とそのISP 80間のリンクになっている。その最低速バンド幅は、事実上の最大バンド幅 (maximum de fact bandwidth) である。

【0012】ここでは、コンテキストから明らかである場合を除き、ネットワークエンティティ（サーバ20とクライアント90のような）間のバンド幅というときは、これらエンティティ間の事実上の最大バンド幅であるものと想定される。

【0013】【0013】バンド幅は、「コネクション速度 (connection speed)」、「速度」、または「レート」とも呼ばれる。本明細書の中でバンド幅というとき、それが毎秒ビット数で表されているときは、「ビット・レート (bit rate)」または「ビットレート (bit rate)」とも呼ばれる（なお、本明細書では「ビットレート」ということにする）。

【0014】ストリーミングメディア

ストリーミング (streaming) は、一定で連続するストリームとして処理できるような形でマルティメディアデータを転送する手法である。ストリーミングテクノロジの重要度がインターネットの成長と共に高

まっているのは、大部分のユーザによるアクセスが十分に高速でないため、大量のマルチメディアファイルを高速にダウンロードできないからである。ストリーミングを使用すると、クライアントブラウザまたはプラグインは、ファイル全体の伝送が完了する前にデータ表示を開始することができる。

【0015】ストリーミングが働くためには、データを受信するクライアント側は、データを収集し、そのデータを一定のストリームとしてアプリケーションに送信し、そのアプリケーションでデータを処理し、それをサウンドやピクチャに変換できるようになっていなければならない。このことは、ストリーミングクライアントが必要以上の高速でデータを受信したとき、余分のデータをバッファに置いておく必要があることを意味する。逆に、データが十分に高速で受信されないとときは、データのプレゼンテーションはスムーズに行われないことになる。

【0016】オーディオおよび／またはビジュアルプレゼンテーションのコンテキスト内では、「メディア」と「マルティメディア」は、本明細書では同じ意味で用いられている。メディアとは、テキスト、グラフィックス、ビデオ、アニメーション、および／またはサウンドを統合化して提示することを意味している。

【0017】「ストリーミングメディア」とは、ネットワーク（インターネットなど）を利用してエンドユーザーに伝送されるオーディオおよび／またはビジュアルプレゼンテーションである。このような伝送は、プレゼンテーションが相対的にスムーズで、ジャーキー (jerk) が生じないよう行われる。追加のフレームがユーザーにダウンロードされている間のポーズ (pause) が長いと、ユーザーにとっては煩わしくなっている。このような煩わしさは、ユーザーが将来のストリーミングメディアを見ることを断念させる原因になっている。

【0018】【0018】ストリーミングメディアのスムーズな伝送
クライアントがデータを受信するときのレートはバンド幅によって決まるので、ストリーミングメディアプレゼンテーションが提示されるときのレートは、バンド幅が許容するレートまでに制限されている。例えば、メディアサーバ20が、ストリーミングメディアプレゼンテーションをスムーズに「プレイ」するためには、データを50 Kbpsでクライアント90に送信する必要があるとする。しかし、クライアントとサーバ間のバンド幅は30 Kbpsに制限されている。その結果は、メディアプレゼンテーションがジャーキーし、ジャンプすることになる。

【0019】この問題を緩和する試みとして、ストリーミングメディアプレゼンテーションは、品質度の異なる複数のフォーマットに符号化（エンコード）されている。

【0020】最低品質のフォーマット（例えば、小サイズ、低解像度、小カラーパケット）は、一定時間にクライアントにpusshされるデータ量が最小になっている。従って、低速リンク上のクライアントはストリーミングメディアプレゼンテーションをスムーズに提示できるが、プレゼンテーションの品質が犠牲になっている。

【0021】最高品質のフォーマット（例えば、フルスクリーンサイズ、高解像度、大カラーサイズ）は、一定時間にクライアントにpusshされるデータ量が最大になっている。従って、高速リンクをもつクライアントは、ストリーミングメディアプレゼンテーションをスムーズに提示できると共に、高品質のプレゼンテーションが得されることになる。

【0022】バンド幅選択アプローチ

サーバはストリーミングメディアをクライアントに送信するとき、どのようなフォーマットを使用するかを知つていなければならない。従って、正しいフォーマットを選択するためには、サーバはサーバとクライアント間のバンド幅を知つていなければならない。

【0023】これを行う最も簡単な方法は、どのようなバンド幅であるかをクライアントのユーザに尋ねることである。インターネットへのクライアントのリンクは、バンド幅にボトルネックがあるのが代表的であるので、このリンクのバンド幅が分かると、実際のバンド幅が分かるのが一般的である。

【0024】図2は、クライアントのコンピュータに表示されるWebページの1部分（カッタウェイ）100を示す。このカッタウェイ100の内部には、そのコネクション速度についてユーザに尋ねるために使用できる代表的なユーザインターフェースがある。ユーザは、ユーザインターフェース110によって提供される、3つのボタン112、114、116をクリックする。ユーザがボタン112をクリックすると、サーバは、28.8 Kbpsで伝送する設計になったフォーマットでストリーミングメディアを収めているファイルからデータを送出する。同様に、ユーザがボタン114をクリックすると、データは、56.6 Kbpsで伝送する設計になったフォーマットでストリーミングメディアを収めているファイルから送信される。ユーザがボタン114をクリックすると、サーバは、56.6 Kbps以上であって、T1コネクションの代表的速度までのレートで伝送する設計になったフォーマットでストリーミングメディアを収めているファイルからデータを送信する。

【0025】しかし、「バンド幅選択」アプローチによると、ユーザに慎重な選択が要求されるという大きな問題がある。このアプローチによると、選択エラーが起こりやすくなっている。

【0026】ユーザはコネクション速度に注意し、理解し、知識があることが要求されている。よくあること

は、ユーザがどのボタンを押すべきかに特別な注意を払わないことである。ユーザが分かっていることは、ユーザがボタンの1つを押すと、メディアプレゼンテーションが表示されることだけである。従って、ユーザはボタンのどれかを押している。

【0027】また、よくあることは、ユーザがバンド幅の概念を理解していないことである。ユーザがボタン116を選択するのは、プレゼンテーションを最高品質で見たいからである。プレゼンテーションを最高品質で表示すると、インターネットコネクションは、インターネットコネクションを通してデータが送信されるときのレートを処理できないので、プレゼンテーションがスムーズでなくなるということがこのユーザに分かっていない。

【0028】ユーザはバンド幅の概念を理解していても、自分のバンド幅が分かっていないことがある。ユーザは自分のバンド幅に無知だけのこともある。さらに、ノイズの度合が変化すると、ユーザがインターネットに接続するたびにコネクション速度が変化することがある。さらに、コネクションのタイプによっては（ケーブルモデムなど）、多数の要因によってコネクション速度が広範囲に変化するものもある。

【0029】さらに、ユーザは、選択が正しくないと、どのようなことが起こるかを理解している必要がある。ユーザは、スムーズなプレゼンテーションを得るために自分のバンド幅と同じか、あるいはそれ以下のオプションを選択する必要があることを理解できるだけの教育が必要である。しかし、ユーザは、自分のバンド幅より大幅に低いオプションを選択してはならない。そのようにすると、利用できる、もっと高いバンド幅で表示できるはずのスムーズなプレゼンテーションが、低品質で表示されることになってしまう。

【0030】以上の説明から理解されるように、このマニュアル（手動操作）による方法は、多くのユーザを混乱させ、脅威になっていることがよくある。そのため、選択を誤ることがよく起こっている。

【0031】さらに、多数のファイル（バンド幅ごとに1ファイル）をメディアサーバに置いておくと、Webサイトを維持するオーバヘッドが増加することになる。

【0032】自動バンド幅検出

上記問題を解消するために、メディアサーバはシングルファイルを使用し、複数のバンド幅用のサブファイルをそこに収めておくことができる。さらに、メディアサーバは、バンド幅を自動的に検出することができる。

【0033】このシングルファイルは、MBR (multiple bit rate: 複数ビットレート) ファイルと呼ばれている。MBRファイルは、複数の異なる「バンド」または「ストリーム」を収めているのが代表的である。これらのバンドは「サブファイル」と呼ばれている。ユーザは1つのリンクをクリックするだけで

ある。すると、サーバは、クライアントに送信するときの正しい速度のバンドを、自動的に判断する。なお、これはユーザには見えないように行われる。

【0034】この自動速度検出は時間がかかることがある。このことは、プレゼンテーションが始まるまでに、ユーザはさらに5秒または1分（またはそれ以上）待たされることを意味している。既存の自動速度検出にこのような遅れがあるのは、速度判断が進行中のときの「ハンドシェーリング」時間が長いことによる。

【0035】ある既存自動速度検出手法では、サーバとクライアント間の速度を測定するために複数のデータパケットが送信されている。この手法は、以下の「複数測定パケット手法」の個所に詳しく説明されている。

【0036】バンド幅測定パケット

代表例として、自動バンド幅検出手法では、既知サイズの1または2以上のパケットを送信することによって、ネットワーク上のエンティティ間のバンド幅を測定している。

【0037】図3は、送信側（例えば、サーバ）と受信側（例えば、クライアント）間の2つのパケット（ P_x と P_y ）の伝送をたどって行く時間グラフを示す。サーバ側とクライアント側はそのように示されている。グラフ上では、時間は下に向かって進んでいる。

【0038】時刻 t_a は、 P_x の伝送が開始されるサーバ側の時刻を示している。時刻 t_b は、 P_x の伝送が終了するサーバ側の時刻を示している。同様に、時刻 t_c は、クライアントが P_x の受信を開始する時刻を示している。時刻 t_d は、クライアントが P_x の受信を完了する時刻を示している。 t_d 時に、ネットワークハードウェアは、通信層を上方に向かってパケットを渡していく、アプリケーション層に到達するものと想定されている。

【0039】パケット P_y は、図3の時間グラフ上に同じように示されている。 t_e は、 P_y の伝送が開始されるサーバ側の時刻である。 t_f は、 P_y の伝送が終了するサーバ側の時刻である。同様に、 t_g は、 P_y の受信を開始するクライアント側の時刻である。 t_h は、 P_y の受信を完了するクライアント側の時刻である。 t_h 時に、ネットワークハードウェアは、通信層を上方に向かってパケットを渡していく、アプリケーション層に到達するものと想定されている。

【0040】シングルパケットを使用したバンド幅測定。研究所のように制御された環境では、ネットワーク上の2エンティティ間のバンド幅測定は簡単である。この計算を行うために、既知サイズのパケットが一方のエンティティから他方のエンティティに送信され、伝送レイテンシが測定されている。ここで、伝送レイテンシ(transmission latency)とは、パケットがソース(発信元)からデスティネーション(宛先)まで走行して行くのに要する時間量のことである。

ある。このようなシナリオが与えられているとき、パケットが送信された時間とパケットが到着した時間が分かっていなければならない。

【0041】この手法は、研究所以外ではほとんど非実用的である。この手法は、クライアントとサーバ間が同期している必要があるので、非同期ネットワーク（インターネットなど）では使用できない。クライアントとサーバのどちらも、同じクロックを使用している必要がある。

【0042】別の手法として、クライアントは、パケットの受信を開始した時刻（ P_x では t_0 ）と、パケットの受信が完了した時刻（ P_x では t_1 ）をたどって行くことができる。

【0043】図3は、サーバからクライアントに送信されるパケット P_x を示している。 P_x は、パケットサイズPSは既知サイズ（ビット数）になっている。バンド幅（bw）の計算式は次の通りである。

【0044】

【数1】

$$bw(P_x) = \frac{PS}{t_1 - t_0}$$

式 1 (シングルパケット)

【0045】この手法は理論的には有効であるが、残念ながら実用には適していない。パケットがいつ初めて受信されるかは、ハードウェアだけが分かっている。従って、 t_0 がいつであるかは、ハードウェアだけが分かっている。

【0046】その他の通信層（トランスポート層とアプリケーション層など）が分かっているのは、パケットがハードウェアによって完全に受信された時刻だけである。これは、ハードウェアがパケットをこれらの層に渡したときである。パケット P_x の完了時刻は t_1 である。ある時点が分かっているだけでは、バンド幅を計算することは不可能である。

【0047】パケットペア。パケットペア（packet pair）と呼ばれる手法は、非同期ネットワークでの上記問題を解消するために使用されている。パケットペアによると、2つの同一パケットはバックツーバック（back-to-back）で送信される。サーバは、一方が他方の直後に続くように、ペアのパケットを送信する。両パケットは同じであるので、同一サイズ（PS）になっている。バンド幅は、パケットサイズを各パケットの受信の時間差で除することによって求められている。

【0048】各パケットは固有の測定可能特性をもっている。具体的には、これらの特性としては、そのパケットサイズ（PS）とパケット到着といった測定時間（例えば、図3中の t_0 ～ t_3 ）がある。ある種の特性（パケットサイズなど）は、測定されるよりも指定され

ことがあるが、測定することが望ましければ、測定することも可能である。

【0049】図3に示すように、サーバはパケット P_x を送信する。クライアント側ハードウェアは t_0 時にパケットの受信を開始する。パケットの受信が t_1 時に完了すると、ハードウェアは通信層を上方に向かってパケットを渡していく。最終的には、パケットはほぼ t_1 時にデスティネーション層（例えば、アプリケーション層）によって受信される。

【0050】サーバは P_x を送信したあと（これは t_b 時に完了している）、 t_c 時にパケット P_y を即時に送信する。重要なことは、1) t_b と t_c の間に測定可能な遅延が絶対に存在しないか、あるいは2) t_b と t_c の間に存在する遅延の長さが分かっていることである。ここでは、説明を簡単にするために、 t_b と t_c の間に測定可能な遅延が存在しないものとしている。

【0051】クライアント側ハードウェアは t_2 時に P_y の受信を開始する。このパケットの受信が t_3 時に完了すると、ハードウェアは通信層の上方に向かってパケットを渡していく。最終的に、パケットはほぼ t_3 時にデスティネーション層（例えば、アプリケーション層）によって受信される。

【0052】図3は、 t_1 （ P_x の受信完了時刻）と t_2 （ P_y の受信開始時刻）の間に遅延が存在しないことを示している。理論的には、このことが常に起こるのは、 P_x と P_y が同じ条件で伝送されたときである。実際には、このことがよく起こるのは、 P_y が P_x の直後に送信されるからである。

【0053】パケットペアを使用したときの、バンド幅（bw）の計算式は次の通りである。

【0054】

【数2】

$$bw(P_x, P_y) = \frac{PS}{t_2 - t_1}$$

式2 (パケットペア)

【0055】この手法は理論的にも、実用面でも有効である。しかし、これが有効に働くのは、比較的静的なネットワークに限られている。

【0056】例えば、図1に示す想定では、ネットワークはサーバ20、ルータ32、34、36、ISP80の中の特定ISP、およびクライアント90から構成されている。さらに、この静的ネットワーク上の各ノード間のリンクは固定され、一定のバンド幅をもっていることが想定されている。このような状況では、パケットペア手法によると、正確で、効果的なバンド幅測定が得られる。

【0057】パケットペアをインターネット上で使用したときの問題。しかし、パケットペア手法は、インターネットのような動的ネットワークでは有効な働きをしな

い。動的ネットワークとは、パケットが先行パケットとは異なるか、後続パケットとは異なる方法で処理される可能性があるようなネットワークである。具体的には、TCPネットワークで問題が起こっている。

【0058】図1は、動的ネットワーク上で見られる差異を処理する例を示している。ここでは、すべてのパケットはサーバからクライアントへ（図1の左から右へ）走行するものと想定されている。また、パケット60～68は、サーバ20によってバックツーバックでクライアントに送信されたものと想定されている。

【0059】図1を見れば明らかのように、パケットは異なるルートを通ることが可能になっている。さらに、ルートによっては、パケット伝送を大幅に遅延させるものがある。このことが起こるのは、特に、パケットが、ワイヤレス伝送のような見かけ上異常な（必ずしも、通常でないとは限らない）ルートを経て伝送され、海底ケーブル、サテライト伝送（ディッシュ46、50とサテライト48で図示）などを経て海外に伝送されるときである。ルータ（ルータ42のような）は、パケットをメモリ（バッファ43のような）に一時的に置いておくことにより、別のルータよりも1または2以上のパケット（63と64のような）を遅延させることができる。

【0060】複数測定パケット手法

上述した問題を解消するために、従来の自動バンド幅測定手法では、複数のパケットが使用されている。サーバは複数のパケット（3以上のパケット）を送信し、各パケットの速度を計算している。バンド幅測定の従来の考え方から明らかであるように、正確な測定を得るために複数のペアのパケットを、数秒間から数分間にわたって繰り返し送信する必要がある。ここでは、上述した「パケットペア」手法と区別するために、この手法は「複数パケット」と呼ぶことにする。

【0061】代表例として、最終的バンド幅は、多数のバンド幅測定の平均を求ることによって判断されている。このように平均をとると、各パケットの遅延のばらつきは平滑化されるが、伝送中のパケット圧縮は補償されない。2測定のうち一方が非常に不正確であると、平均はゆがむことになる。

【0062】残念ながら、この手法は、クリックしてからメディアプレゼンテーションまでユーザが待たされていた時間に比べて待ち時間が長くなっている。この待ち時間は、データと状況に応じて5秒から数分間にわたっている。このような待ちがあると、メディアプレゼンテーションを体験することを望んでいるユーザにとって煩わしが増加することになる。この遅延は望ましくない。従来の手法では、利用できる他のオプションがないために、ユーザはこれらの遅延に我慢せざるを得なくなっている。

【0063】どの既存自動バンド幅測定手法も、インターネット上のバンド幅を、ペアのパケットを使用してほ

ば瞬時に測定することが不可能である。また、どの既存自動バンド幅測定手法も、この測定をアプリケーション層で行うことが不可能である。そのために、オペレーティングシステムを修正することが回避されている。どの既存自動バンド幅測定手法も、パケット圧縮が原因で起こる測定ひずみを扱っていない。

【0064】トランスポート層の実装

従来のアプローチでは、自動バンド幅測定を行うためにオペレーティングシステム（OS）のカーネルを修正しているのが代表的である。具体的に説明すると、これらのアプローチでは、OS I モデルのトランスポート層を修正しているが、この層はOSのカーネルに置かれていることが多い。一般的に、このような修正は、OSを修正していない実装に比べて、全体的に安定性が低下し、高価になるので望ましくない。

【0065】これらのアプローチはアプリケーション内に（従って、アプリケーション層に）実装できないので、このような修正是不可能になる。しかし、どの既存パケットペアによる手法も、アプリケーション層でバンド幅を測定していない。これは、ネットワーク上で実際に行われるコミュニケーション詳細に対する制御権が、アプリケーション層では、他の層に比べて低いためである。具体的には、アプリケーションは、TCP使用による制御権が、UDP（User Datagram Protocol：ユーザデータグラムプロトコル）による場合よりも、はるかに低くなっている。

【0066】TCPとUDPは、以下の「TCPとUDP」の個所に詳しく説明されている。トランスポート層とアプリケーション層は、以下で説明するOS I モデルの7層の一部になっている。

【0067】TCPとUDP

インターネット（および他のネットワーク）上では、データパケットは、TCPまたはUDPプロトコルを使用して送信されるのが通常である。TCPはインターネットでは広く認められ、理解されている。

【0068】TCP（Transmission Control Protocol：伝送制御プロトコル）は、TCP/IPネットワーク（インターネットなど）における主要プロトコルの1つである。IPプロトコルはパケットだけを扱っているのに対し、TCPは2つのホストがコネクションを確立し、データストリームをやりとりすることを可能にしている。TCPは、データの配達を保証し、パケットがその送信時と同じ順序で配達されることを保証している。

【0069】UDP（User Datagram Protocol：ユーザデータグラムプロトコル）はコネクションレス型プロトコル（connection less protocol）であり、（TCPと同じように）IPネットワークの上で走行する。TCP/IPとは異なり、UDP/IPは提供するエラー回復サ

ービスが非常に少ないが、その代わりに、IPネットワーク上のパケット（つまり、データグラム）の送受信を直接的方法で行うことができる。

【0070】パケットは、アプリケーションプログラムによって提供されるデータの塊（chunk）である。UDPでは、シングル「アプリケーションレベルデータ」はシングル UDPパケットとして送信されるのが代表的である。これに対し、TCPでは、シングルアプリケーションレベルパケットは、もっと小さな、複数のTCP「セグメント」に分割され、その各々はTCP層で別々の「パケット」として扱われるようになっている。Nagleアルゴリズム（下述する）は、上記とは逆のことを行う。つまり、複数の、小さなアプリケーションパケットを受け取り、それらを結合してもっと大きな、1つのTCPセグメントにする。

【0071】Nagle TCP/IP アルゴリズム
Nagleアルゴリズムは、低速ネットワーク上で小さなTCPセグメント（「タイニグラム（tinygram）」と呼ばれることもある）に起る問題を回避する設計になっている。このアルゴリズムの規定によれば、まだ受信確認（ACK）されていない未処理のまま、TCP/IPコネクションに残っているタイニグラムは1つに限られている。タイニグラムの定義サイズは実装によって決まる。しかし、そのサイズは、一般的に、代表的なTCPセグメントのサイズより小さくなっている。

【0072】Nagleアルゴリズムのよれば、状況によっては、データが送信される前の待ち時間は約200ミリ秒（ms）であることが規定されている。Nagleアルゴリズムは、スイッチ経由のトラフィックに対して次のようなパラメータを使用している。

- ・ セグメントサイズ = MTUまたはtcp_msdfiltまたはMTU経路検出値
- ・ TCPウィンドウサイズ = tcp_sendspace値とtcp_recvspace値のうち小さい方
- ・ データサイズ = アプリケーションデータバッファサイズ

【0073】以下は、データをいつ送信するかを判断するときNagleアルゴリズムによって使用される、具体的ルールである。

- ・ パケットがセグメントサイズと同じか、それより大きく、TCPウィンドウが一杯にならなければ、MTUサイズのバッファを即時に送信する。
- ・ インタフェースがアイドル中であるか、TCP_NODELAYフラグがセットされていて、TCPウィンドウが一杯にならなければ、バッファを即時に送信する。
- ・ TCP ウィンドウに未処理のまま残っているデータ

がウィンドウの半分以下であれば、バッファを即時に送信する。

- 送信するセグメントサイズがバッファ未満であり、ウィンドウに未処理のまま残っているデータがウィンドウの半分以上あって、TCP_NODELAYがセットされていなければ、残りデータについて200 ms ecまで待ってからバッファを送信する。

【0074】送信側のソケットでTCP_NODELAYをセットすると、Nagleアルゴリズムは非アクチベートされる。送信される全データは、データサイズに関係なく即時に送出される。

【0075】Nagleアルゴリズムは、広い意味では、「タイニグラムバッファリング」機能と呼ぶことができるが、これはタイニグラムがバッファリングされるからである。

【0076】TCP Slow Startアルゴリズム

“Slow Start”（スロースタート）を使用しないTCPネットワークでは、デバイスは、受信側が公表しているウィンドウサイズに達するまで、複数のパケットをネットワークに投入することによって送信側とのコネクションを開始している。2ホストが同じLAN上にあるときは、これは許されるが、送信側と受信側の間にルータや低速リンクがあると、問題が発生する可能性がある。中間ルータの一部はパケットを待ち行列（キューリ）に置いておく可能性があるので、そのようなルータは、メモリ不足のためパケットを待ち行列に置くことができないことが起こり得る。従って、このナイーブなアプローチによると、TCPコネクションのスループットが大幅に低下するおそれがある。

【0077】これを回避するアルゴリズムは“Slow Start”と呼ばれている。このアルゴリズムは、新パケットがネットワークに投入されるときのレートが、受信確認通知が他方のエンドから戻されるときのレートであることを観察する働きをする。

【0078】このSlow Startアルゴリズムでは、送信側のTCPに別のウィンドウが追加されている。つまり、輻輳（congestion）ウィンドウであり、“cwnd”と名付けられている。別のネットワーク上のホストとの間で新しいコネクションが確立されると、輻輳ウィンドウは1パケットに初期化される。受信確認通知（つまり、“ACK”）が受信されたたびに、輻輳ウィンドウは1パケットずつ増加して行く。送信側は、「輻輳ウィンドウ」と「公表（advertised）ウィンドウ」の最小限まで送信することができる。「輻輳ウィンドウ」は、送信側によって課されたフロー制御である。「公表ウィンドウ」は、受信側によって課されたフロー制御である。前者は、ネットワーク輻輳がどの程度であるかを送信側が評価することに基づいている。後者は、そのコネクションのために利用

できるバッファスペースが、受信側にどれだけ残っているかに関係している。

【0079】送信側は、1パケットを送信し、そのACK（受信確認通知）を待つことからスタートする。そのACKが受信されると、輻輳ウィンドウは1から2にインクリメントされる。これらの2パケットの各々が受信確認されると、輻輳ウィンドウは4に増加する。以下、同様である。

【0080】いずれかの時点で、送信側と受信側間のコネクションの容量まで到達することになる。容量まで到達すると、いずれかの中間ルータはパケットを破棄することを開始する。輻輳ウィンドウが限界まで達したことは、この廃棄によって送信側は知ることになる。

【0081】プロキシ

プロキシ（proxy）（つまり、プロキシサーバ）は、クライアントアプリケーション（Webブラウザなど）と実サーバ（real server）の間に置かれたデバイスである。一般的に、プロキシは、実サーバとの間でやりとりされるすべての要求をインターフェース（intercept）し、自身で要求を満たすことができるかどうかを確かめる。満たすことができなければ、要求を実サーバに転送する。プロキシは、主に次の2目的のために採用されている。すなわち、パフォーマンス向上と要求のフィルタリングである。

【0082】プロキシは、複数のクライアントのための中央コミュニケーションポイントとなることが多いので、そのコミュニケーションを可能な限り効率化することを試みている。従って、プロキシは、Nagleアルゴリズムの一種を実装しているのが代表的である。すべての新コネクションはSlow Startから開始する。クライアントとサーバの間にプロキシが置かれているときは、SlowStartは、2つのコネクションで実行される。すなわち、サーバとプロキシ（server-proxy）間コネクションとプロキシとクライアント（proxy-client）間コネクションである。従って、プロキシを使用すると、パケットペアの試みはさらに複雑化することになる。

【0083】

【発明が解決しようとする課題】背景の要約

アプリケーション（アプリケーション層に置かれている）は、TCPパケットの処理に対する制御が制限されている。従って、従来のバンド幅測定では、アプリケーションレベルでのTCPバンド幅測定が回避されている。

【0084】パケットペア手法では、少なくとも2つのパケットはバックツーバックで送信されることが保全性の要件となっている。しかし、これらのパケットは、NagleアルゴリズムとSlow Startアルゴリズムの影響のためにそのような形で到着しないことがある。このことが、TCPネットワーク上のバンド幅

測定にパケットペア手法を使用することの妨げとなる。いる。

【0085】そこで、本発明の目的は、パケットペア手法を使用することの可能なTCPネットワーク環境におけるバンド幅の高速動的測定方法を提供することにある。

【0086】

【課題を解決するための手段】概要

TCPネットワーク環境における高速動的バンド幅測定では、ネットワーク（インターネットなど）上の2エンティティ間のバンド幅を計算するために、シングルペアのパケットが利用されている。この計算はパケットペア手法に基づいている。このバンド幅測定は非常に高速である。

【0087】パケットがネットワークを走行している途中で、通信デバイスはパケットペアを遅延させことがある。具体的には、TCPネットワークでは、ネットワークの総スループットを向上することを目標として、ある種のパケットを遅延させる設計になった2つのアルゴリズムが使用されている。しかし、これらのアルゴリズムは、バンド幅を測定することを目的としたパケットペアを事実上遅延させる可能性がある。従って、これらのアルゴリズムを使用すると、測定にひずみが起こっている。そのアルゴリズムとは、"Nagle"と"Slow Start"である。

【0088】高速動的バンド幅測定では、これらのアルゴリズムに課されている遅延を解消するための対応策が実装されている。その対応策とは、Nagleアルゴリズムの適用を禁止すること、「プッシュ」パケットをパケットペアの直後に送信することによってパケットのバッファリングを最小限にすること、およびダミーパケットで初期化（プライミング）することによってSlow Startアルゴリズムを回避することである。

【0089】

【発明の実施の形態】以下では、TCPネットワーク環境におけるバンド幅の高速動的測定の具体的実施形態について説明するが、そこには請求項に記載されているエレメントが組み込まれている。この実施形態は、法律上の要件、使用可能性の要件、および最良形態の要件に合致するように具体的に説明されている。しかし、その説明自体は、本発明の範囲を限定するものではない。むしろ、本発明によるTCPネットワーク環境における高速動的バンド幅測定は、他の現存および将来のテクノロジと併用して他の方法でも実現可能であり、それが本発明者の意図する目的である。

【0090】例示のTCPネットワーク環境におけるバンド幅の高速動的測定（つまり、「バンド幅メータ」または「bw-meter」）。なお、以下では、「バンド幅メータ」という）はTCPネットワーク（インターネットなど）で使用されたときでも、高速であり、強固で

ある。例示のバンド幅メータは、通信ネットワーク上の2エンティティ間で利用可能なネットワークバンド幅を自動的に測定する低レイテンシ手法が実装されている。これは、インターネット（または他のTCPネットワーク）上で利用すると、特に有用であることが判明している。

【0091】従来のアプローチとは異なり、例示のバンド幅メータによると、困難なネットワーク条件下でも、起こり得る遅延を最小にして最善努力のバンド幅測定が得られる。例示のバンド幅メータは、LAN、ケーブル、DSL、およびモdemコネクションを含む、大部分の既存TCPネットワークにおいて1秒未満で妥当な出力が得られる設計になっている。

【0092】さらに、例示のバンド幅メータは、アプリケーション層に実装されている。例示のバンド幅メータは他の層でも実装可能であるが、ここで説明されているものはアプリケーション層に実装されている。具体的には、これは、その一部をWebブラウザまたはメディアプレイヤで実現することが可能である。

【0093】例示のバンド幅メータによって実現可能なパケットペア手法の他の側面は、米国での特許出願、発明の名称「コネクションバンド幅の高速動的測定（Fast Dynamic Measurement of Connection Bandwidth）」、Microsoft Corporationに譲渡済み）に詳しく説明されている。なお、この米国特許出願の内容は引用により、本明細書の一部になっている。

【0094】パケットペア手法

例示のバンド幅メータでは、上述し、図3に示されている、確立されたパケットペア手法が利用されている。例示のバンド幅メータは、上述したパケットペア公式（式2）を使用して、通信ネットワーク（インターネットなど）上の2エンティティ間の、事実上の最大バンド幅を計算している。

【0095】複数のパケットを使用している、既存自動バンド幅測定手法とは異なり、例示のバンド幅メータは、インターネット上のバンド幅を測定するためにシングルペアのパケットを使用している。例示のバンド幅メータでは、バンド幅測定と計算は、シングルペアの測定パケットだけが送信されるので、「ほぼ瞬時」に行われる。ここで「ほぼ瞬時に（nearly instantaneously）」とは、ペアのパケットがクライアントに到着するのと同時にバンド幅が求められることを意味している。

【0096】例示のバンド幅メータは、NagleアルゴリズムとSlow Startアルゴリズムに対する対応策を実装することによって、TCPネットワーク（インターネットなど）上でパケットペアを使用するときの欠点と制約を解消している。

【0097】パケットペアが通るルート。例示のバンド

幅メータのパケットペア手法によるパケットは、送信側エンティティ（例えば、サーバ）から受信側エンティティ（例えば、クライアント）に走行する。図4は、そのような走行ルートの例を示す。図4に示す環境は、図1に示す環境に類似している。

【0098】図4は、代表的なインターネット（TCPネットワーク）構成の例を示す図である。そこには、サーバ（メディアサーバ220など）が含まれ、これはインターネット230に結合されている。サーバは、1または2以上の物理的サーバコンピュータ222が実装され、コンピュータは1または2以上の物理的ストレージデバイスおよび／またはデータベース224を装備している。インターネット伝送の反対側には、クライアント290、292が置かれており、これは、インターネットサービスプロバイダ（Internet Service Provider ISP）280である、プロキシサーバ284を経由して接続されている。

【0099】インターネットはクラウド230で示されているが、理解されるように、このクラウドは、そこに示されているものだけが置かれているインターネット部分を表している。このクラウドの内側には、ルータ、伝送回線、コネクション、および他のデバイスが置かれ、クライアントとサーバ間のデータ伝送が、ほぼ正常に行われるようになっている。例示のインターネットクラウド230の内側には、ルータ232～244、2つのサテライトディッシュ246、250、およびサテライト248が置かれている。これらは、データパケットがサーバとクライアント間を走行する途中で取り得る経路を表している。

【0100】図4は、例示のバンド幅メータに従って送信されるペアの、連続するパケット260、262を示している。サーバ220は、パケット260の直後にパケット262を送信する。

【0101】プロキシサーバ284は、リンク282を通してそのISP280に接続されている。クライアント290とクライアント292はプロキシサーバを使用して、インターネットと通信する。

【0102】アプリケーションレベルでのバンド幅測定
従来のバンド幅測定アプローチは、アプリケーション層の下のトランスポート層または他の、なんらかの層に実装されているのが代表的である。しかるに、例示のバンド幅メータはアプリケーション層に実装されている。アプリケーションレベルでTCPパケットペアバンド幅を測定する方法には、少なくとも2つの大きな利点がある。

【0103】第一は、下位レベル（トランスポート層など）にパケットペア手法を実装することは、不利であることである。OSのカーネルの変更が必要になり、段階的（増分的）展開に役立たないからである。アプリケーションレベル実装とは異なり、下位パケットペア実装は

その開発、初期展開、将来の開発、および将来の展開に多大な費用を要している。

【0104】第二は、ある研究によれば、バンド幅測定の恩恵を受けるのは、研究対象のTCPコネクションのうち1/4にすぎないことがある。従って、使用されるコネクションがその1/4未満にすぎないときは、このバンド幅測定を下位レベルに実装することはコスト効率的でない。従って、このバンド幅は、それを必要とするアプリケーションに組み込んでおくのが最良である。アプリケーションは、オペレーティングシステムの新しいカーネルよりも、段階的（増分的）展開がはるかに容易である（低コストである）。

【0105】一般的に、受信側クロックが十分な精度であり、IPデータグラム（つまり、パケット）が受信側ネットワークスタック（OS層）を上方に向かって通り抜けてアプリケーションに妨害を受けることなく渡されるようになる考え方が安全な方法である。America Onlineバージョン4（AOLv4）およびそれ以前のソフトウェアのように、タイマだけでデータがアプリケーションに渡されるように振る舞っていた、ある種のアプリケーションの場合には、この考え方に対することになる。従って、測定には人工的クロックの細粒性が要求されていた。幸いなことに、AOLソフトウェアのバージョン5（およびそれ以降）では、このような妨害は起こっていない。

【0106】パケットペア使用による効果的測定の条件
パケットペア手法を使用してバンド幅を測定するとき、良好な測定を得るためにには2つの条件を満足しなければならない。

【0107】第一の条件は、パケットがバックツーバックで送信されなければならないことである。ここでは、これを「バックツーバック」条件と呼ぶことにする。パケットがバックツーバックで送信されないときは、パケット間のタイミング測定にひずみが生じることになる。NagleアルゴリズムとSlow Startアルゴリズムはどちらも、この条件に反するおそれがある。どちらも、2番目の測定パケットの配達を遅延させる潜在性をもっている。パケットペアを使用してバンド幅を測定する場合は、実際のバンド幅の測定にひずみが生じるので、パケット間に起こることが避けられない遅延は耐えられないものになっている。

【0108】第二の条件は、パケットのサイズは保たれていなければならないことである。つまり、パケットは他のパケットと結合してはならない。ここでは、これを「サイズ保持」条件と呼ぶことにする。Nagleアルゴリズムは、この条件に反するおそれがある。

【0109】Nagleアルゴリズムによると、複数のアプリケーション層パケットはシングルTCPパケットとして送信されることがある。従って、アプリケーションは2またはそれ以上のパケットを送信しているつも

りでいても、実際には、TCP層はシングルパケットだけを送信している。

【0110】Nagle アルゴリズムに対する対応策
Nagle アルゴリズムの興味のある振る舞いは、小さなパケットのとき、未処理のまま残っているACKは1つだけであることである。従って、ペアになった小さなパケットは、Nagle アルゴリズムではバックツーバックで送信することができない。Nagle アルゴリズムでは、ACKを待っている小さなパケットは結合されることになる。これは、「バックツーバック」条件にも、「サイズ保持」条件にも影響を与えている。

【0111】例示のバンド幅メータでは、Nagle アルゴリズムがこれら2条件を妨げる傾向を解消するための対応策が講じられている。あるエンティティ(図4中のサーバ220など)はコマンドを送信し、通信デバイス(ルータ230-250など)がNagle アルゴリズムを使用禁止(disable)にするよう指示する。一般的に、サーバは「ディレイディスエーブル(delay-disable)」コマンドと総称されるコマンドを渡している。具体的には、サーバはTCP_NODELAYをSetSockOpt()に渡している。

【0112】輻輳ウィンドウがオープンしている限り、Nagle アルゴリズムをオフにすると、TCPはpacket-pairパケットのいずれも結合する試みが禁止されるので、TCPは同時にパケットをネットワークに書き出すことになる。

【0113】言い換えれば、Nagle アルゴリズムを「ディレイディスエーブル」コマンドで使用禁止にすると、packet-pairのどちらのパケットも、Nagleが複数パケットを収集するとそれが原因で起こる遅延なしで、ルータを通過していくことになる。

【0114】Slow Start アルゴリズムに対する対応策

例示のバンド幅メータでは、Slow Start アルゴリズムが「バックツーバック」条件を妨げる傾向を解消するための対応策が講じられている。これは、サーバの輻輳ウィンドウ(これは具体的には" cwnd"と名付けられている)を少なくとも3パケットにオープンすることによって行われる。

【0115】これは、輻輳ウィンドウを「プライミング(priming)」(初期化)することによって行われる。輻輳ウィンドウをプライミング(初期化)するために、サーバは少なくとも1つのパケットを送信し、ACKを受信してから、packet-pairのペアのパケットを送信するようにしている。従って、サーバは、少なくとも1つの「プライミング」パケットをクライアントに送信するが、そのパケットはバンド幅計算のために使用されない。1つまたは2つ以上のプライミングパケットが送信されたあと、サーバは、バンド幅計

算のために実際に使用されるpacket-pairを送信する。この時点では、Slow Start アルゴリズムでは、少なくとも2パケットが遅延することなく、続けて通り抜けるようにしている。

【0116】Slow Start アルゴリズムは、特定のTCPコネクションの後半でバンド幅測定を行うようにすると、完全に回避することができる。しかし、これは、2つの理由で望ましいオプションではない。遅延とオーバヘッドが増加し、誤った測定の原因となるからである。

【0117】測定が後半で行われる場合は、組み込まれた遅延があるため、Slow Start アルゴリズムが全コースを走行するまで待たされることになる。回避できる遅延ならば、ない方がよい。例示のバンド幅メータによると、この遅延を回避することができる。

【0118】バンド幅測定をTCPコネクションの先頭で行うと、コネクションの進行と共に累積して行く、多数の不確実性が除去される。例えば、TCPコネクションが制御層とデータトランスポート層の両方で共有されているときは、送信側の輻輳ウィンドウにパケットがパケットツーバックで送信される余裕があるかどうかを、セッションの後半で予測することは不可能である。

【0119】プロキシでの遅延に対する対応策

プロキシで動作しているNagle アルゴリズムは、同じように、packet-pairのバンド幅測定にひずみを生じさせる可能性がある。一般的に、プロキシは「ディレイディスエーブル」コマンドを認識しない。クライアントとサーバのどちらのアプリケーションも、コネクションが回線レベルのプロキシを通るように行われているかどうかを、事前に通知することができない。

【0120】プロキシに置かれたNagle アルゴリズムにアドレスするためには、ペアの測定パケットのあとに大きな第3パケットが送信されている。プロキシがpacket-pairの第2パケットを保持していると、この第3パケットは第2パケットをプッシュ(押し出す)ことになる。従って、この第3パケットは「プッシュ(push)」パケットと呼ばれている。

【0121】さらに、第1パケットと第2パケットは、プロキシで結合することが可能である。このようにすると、人工的に得られる測定は高くなるが、いずれにしても、プロキシユーザの圧倒的多数は高バンド幅コネクションをもつことになる。

【0122】方法による実装

図5は、例示のバンド幅メータを方法で実現した例を示す。これは、サーバ側から見た図である。300で、例示のバンド幅メータによる動的バンド幅測定が開始される。代表例として、クライアントのユーザは、Webページ上でオプションを選択すると、メディアプレゼンテーションを体験することになる。別の方法として、このバンド幅測定を、クライアントに置かれたアプリケーシ

ヨンで開始することも可能である。そのようなアプリケーションとしては、Webブラウザ、メディアプレイヤなどがある。

【0123】一般的に、図5の302で、サーバは、一方が他方の直後に続くように、ペアのパケットをクライアントに送信する。このブロック302での具体的な実装詳細は、図6、図7、および8に示されている。これらの図については、下述する。

【0124】306で、サーバはクライアントからの応答を待っている。時間制限内に応答が受信されなければ、このプロセスは元に戻り、302で別のペアのパケットが送信される。フローチャートには図示されていないが、このプロセスによれば、これは一定の回数繰り返されたあと、終了し、エラーを生成する。応答が時間制限内に受信されたときは、プロセスは308で次のブロックに進む。

【0125】応答には、サーバから送られてきたペアのパケットを使用してクライアントによって、304で判断されたバンド幅測定が含まれている。308で、サーバは、応答から特定のバンド幅を抽出する。

【0126】図9の310で、サーバは、特定のバンド幅に等しいか、それ以下のバンド幅用にフォーマットされたファイル（またはその一部）を選択する。312で、サーバはそのファイル（またはその一部）をクライアントに送信する。

【0127】それがメディアファイルであれば、クライアントのユーザは、プレイが即時に開始されるメディアプレゼンテーションを楽しむことができる。プレイは、測定されたバンド幅で可能な限り最高品質でスムーズに行われる。プロセスは314で終了する。

【0128】Nagleアルゴリズムに対する対応策。図6は、Nagleアルゴリズムに対する対応策として、例示のバンド幅メータを方法で実現した具体例を示す。402で、サーバは「ディレイディスエーブル」コマンドを送信し、Nagleアルゴリズムの使用を禁止する。404で、サーバはペアのバンド幅測定パケットをクライアントに送信する。406で、プロセスは図5のブロック306に戻る。

【0129】プロキシ遅延に対する対応策。図7は、プロキシ遅延に対する対応策として、例示のバンド幅メータを方法で実現した具体例を示す。412で、サーバはペアのバンド幅測定パケットをクライアントに送信する。414で、サーバは「プッシュ」パケットを送信し、通信デバイスによってパケットがストアされている可能性のあるバッファからペアを押し出す。416で、プロセスは図5のブロック306に戻る。

【0130】Slow Startアルゴリズムに対

する対応策。図8は、Slow Startアルゴリズムに対する対応策として、例示のバンド幅メータを方法で実現した具体例を示す。422で、サーバは「プライミング」パケットを送信し、Slow Startアルゴリズムを解消する。

【0131】この「プライミング」パケットはバンド幅測定には使用されない。このパケットはネットワークをオープンさせ（つまり、輻輳ウィンドウをオープンさせる）、同時に2つのパケットを遅延なしで可能にする。424で、サーバはペアのバンド幅測定パケットをクライアントに送信する。426で、プロセスは図5のブロック306に戻る。

【0132】その他の実装の詳細

実装アプリケーション。例示のバンド幅メータは、ネットワーク上の2エンティティ間のバンド幅を高速に測定することを望んでいるエンティティによって実装されることが可能である。具体的には、ネットワークは、インターネットなどのTCPネットワークである。

【0133】そのようなエンティティは、この例示のバンド幅メータをアプリケーション層に実装することができる。この例示のバンド幅メータを実装できるアプリケーションレベルのプログラムモジュールの例としては、サーバ側に置かれていて、Microsoft Media Server (MMS) プロトコルまたはReal Time Streaming Protocol (RTSP) のどちらかを使用するストリーミングメディアサーバアプリケーションがある。

【0134】MMSとRTSPのどちらも、非常に類似した基本的手法を共有し、例示のバンド幅メータを使用して正常な測定を行うための条件を示している。しかし、RTSPを使用した例示のバンド幅メータの実装は、MMSプロトコルを使用した実装よりも扱いづらくなっている。

【0135】RTSPパケットペアの構文。RTSPをMMSよりも扱いづらくしている1つは、3つのパケットをRTSPコマンドに対する応答であるかのように見せて、クライアントのRTSPバーサがそれらのパケットを処理できるようにする必要があるからである。RTSP GET_PARAMETERコマンドは、パケットペアの実験を要求するために使用される。応答の第1パケットの先頭には、代表的なRTSP応答ヘッダが置かれている。

【0136】次に示したものは、クライアントからのパケットペア要求のヘッダの例である。

【0137】

【数3】

```

GET_PARAMETER * RTSP/1.0
Content-Type: application/x-rtsp-packetpair
Content-Length: 16
Date: Sun, 02 Apr 2000 22:38:18 GMT
CSeq: 2
User-Agent: WMPlayer/5.0.0.0488 guid/1A21De80-08E7-11D4-83FE-
006097B76A2E
Accept-Language: en-us, *,q=0.1
Accept-Charset: UTF-8, *,q=0.1
Timestamp: 1

```

【0138】次に示したものは、サーバからのパケットペア応答のヘッダの例である。

【0139】

【数4】

```

RTSP/1.0 200 OK
Content-Type: application/x-rtsp-packetpair
Content-Length: 2048
Date: Sun, 02 Apr 2000 22:30:48 GMT
CSeq: 2
TimeStamp: 1 0.063
Server: WMServer/5.0.0.0518

```

【0140】TCPの問題。前述したように、輻輳ウィンドウは、少なくとも3パケットがサーバから送られる時点までにその3パケットのためにオーブンしている必要がある。初期輻輳ウィンドウは2であるので、3またはそれ以上のパケットのためにウィンドウをオーブンするためにはDESCRIBE応答が使用される。DESCRIBE応答が3パケットを要求していれば、これは、3番目のパケットはクライアントからのACKを待ってからでなければ送信できないことを意味する。

【0141】サーバのTCPが最初の2パケットの一方または両方のACKを待っている間に、GET_PARAMETERが到着し、アプリケーションがGET_PARAMETERに対する応答をソケットに書くことを始めたときは、パケットペアのパケットはDESCRIBE応答の3番目である最終パケットと結合され、相互に結合されることになる。従って、クライアントは、DESCRIBE応答が完全な形で受信されるまではGET_PARAMETERを送信してはならない。

【0142】このようにすると、パケットペアのパケットが送信されるとき輻輳ウィンドウがサーバ側でオーブンすることが保証される。その結果、どのパケットも結合されることはない。DESCRIBE応答は1またはそれ以上のパケットのことがあるので、輻輳ウィンドウは、パケットペアの実行時に3またはそれ以上になる。パケットペアの前に他のトラフィックが行われることがないことは、もちろんである。

【0143】到着時間の測定。（アプリケーションレベルで）例示のバンド幅メータのパケットペア測定を実行するということは、クライアントアプリケーションが2パケットの到着時間を測定することを意味している。RTSPでは、正確な測定のために要求される細粒性に比べて応答ヘッダはその処理に時間がかかるため、これは余計な難題になっている。従って、クライアントは、応

答の処理が終わるのを待ってからでなければ、パケットペアの第1パケットがタイムスタンプされる前にそれがパケットペア要求に対する応答であることを知ることができない。

【0144】このタイムスタンプは、クライアントがどのタイプの応答であるかを知る前に行われていなければならない。従って、クライアントがパケットペア要求を行うとき、着信するすべてのコマンド応答はパケットペアの受信前にタイムスタンプされる。そのあと、このプレタイムスタンプモード（pre-timestamp mode）から出ることになる。

【0145】このモードから出ても、クライアントは、第1パケットのヘッダを処理してからでなければ第2パケットを読み取ることができない。従って、ボトルネックをどの高さまで測定できるかには上限があり、これは、クライアントがRTSP応答ヘッダをどれだけ高速に処理できるかによって決まる。例えば、ヘッダを処理するのに要する時間が5msであれば、測定できる最大速度は800 kb/s前後である。従って、ハイエンドにおけるRTSP測定は、RTSP応答をパース（parse：構文解析）するのに要する時間が低くなければ、MMSほどには良好にならない。

【0146】例示のコンピューティング環境

図9は、例示のバンド幅メータを実現するのに適しているコンピューティング環境920の例を示す。

【0147】例示のコンピューティング環境920は、適したコンピューティング環境の一例にすぎず、例示のバンド幅メータが使用される範囲または機能する範囲が限定されることを意味するものではない。また、コンピューティング環境920は、例示のコンピューティング環境920に図示のコンポーネントまたはその組み合わせに関して依存性または要求条件があるものと解釈してはならない。

【0148】例示のバンド幅メータは、他の多数の汎用または専用コンピューティングシステム環境または構成で動作可能になっている。例示のバンド幅メータで使用するのに適している周知のコンピューティングシステム、環境、および／または構成の例としては、パーソナルコンピュータ、サーバコンピュータ、シンクライアント（thin client）、シッククライアント（thick client）、ハンドヘルドまたはラップトップデバイス、マルチプロセッサシステム、マイクロプロセッサベースシステム、セットトップボック

ス (set top box)、プログラムブルコンシューマエレクトロニクス、ワイヤレス電話、ワイヤレス通信デバイス、ネットワークPC、ミニコンピュータ、メインフレームコンピュータ、上述したシステムまたはデバイスのいずれかを含んでいる分散コンピューティング環境、などがあるが、これらに限定されるものではない。

【0149】例示のバンド幅メータは、プログラムモジュールのように、コンピュータによって実行されるコンピュータ実行可能命令という広い意味でとらえて説明することもできる。一般的に、プログラムモジュールには、ルーチン、プログラム、オブジェクト、コンポーネント、データ構造などがあり、これらは特定のタスクを実行し、あるいは特定の抽象データ型を実装している。例示のバンド幅メータは、通信ネットワークを通してリンクされているリモート処理デバイスによってタスクが実行されるような分散コンピューティング環境で実施することも可能である。分散コンピューティング環境では、プログラムモジュールは、メモリストレージデバイスを含む、ローカルとリモートの両方のコンピュータ記憶媒体に置いておくことができる。

【0150】図9に示すように、コンピューティング環境920は、汎用コンピューティングデバイスをコンピュータ930の形で含んでいる。コンピュータ920のコンポーネントとしては、1または2以上のプロセッサまたは処理ユニット932、システムメモリ934、およびシステムメモリ934を含む、種々のシステムコンポーネントをプロセッサ932に結合するバス936があるが、これらに限定されるものではない。

【0151】バス936は、数種タイプのバス構造のいずれかの1つまたは2つ以上を表しているが、そのようなものとしては、メモリバスまたはメモリコントローラ、ペリフェラルバス、高速グラフィックスバス、およびプロセッサまたはローカルバスがあり、これらは種々バスアーキテクチャのどれかを採用している。そのようなアーキテクチャの例をいくつか挙げると、ISA (Industry Standard Architecture: 業界標準アーキテクチャ) バス、MCA (Micro Channel Architecture: マイクロチャネルアーキテクチャ) バス、EISA (Enhanced ISA: 拡張ISA) バス、VESA (Video Electronics Standards Association) ローカルバス、およびMezzanineバスとしても知られているPCI (Peripheral Component Interconnects: ペリフェラルコンポーネント相互接続) バスがあるが、これらに限定されない。

【0152】コンピュータ930は、種々のコンピュータ読取可能媒体を装備しているのが代表的である。そのような媒体としては、コンピュータ930によってアク

セス可能ならば、どの利用可能な媒体にすることも可能であり、その中には、揮発性媒体と不揮発性媒体、取り外し可能媒体と取り外し不能(固定)媒体が含まれている。

【0153】図9に示すように、システムメモリに装備されているコンピュータ読取可能媒体は、ランダムアクセスメモリ (random access memory RAM) 940などの揮発性、および/またはリードオンリメモリ (read only memory ROM) 938などの不揮発性の形体になっている。スタートアップ時のときのように、コンピュータ930内のエレメント間で情報を転送するのを支援する基本ルーチンからなる、基本入出力システム (basic input/output system) 942は、ROM 938に格納されている。RAM 940は、プロセッサ932によって即時にアクセス可能である、および/または現在操作可能であるデータおよび/またはプログラムモジュールを格納しているのが代表的である。

【0154】コンピュータ930は、さらに、他の取り外し可能/取り外し不能、揮発性/不揮発性コンピュータ記憶媒体を装備することも可能である。図9には、取り外し不能、不揮発性磁気媒体との間で読み書きするハードディスクドライブ944 (これは、図には示されていないが、「ハードドライブ」とも呼ばれている)、取り外し可能、不揮発性磁気ディスク948 (例えば、「フロッピディスク」との間で読み書きする磁気ディスクドライブ946、およびCD-ROM、DVD-R OM、その他の光媒体などの取り外し可能、不揮発性光ディスク952との間で読み書きする光ディスクドライブ950が示されているが、これらは一例にすぎない。ハードディスクドライブ944、磁気ディスクドライブ946、および光ディスクドライブ950は、それぞれ1または2以上のインターフェースを通してバス936に接続されている。

【0155】これらのドライブおよびその関連コンピュータ読取可能媒体は、コンピュータ読取可能命令、データ構造、プログラムモジュール、および他のデータを、コンピュータ930のために永続的(不揮発性)に保存している。ここで説明している例示の環境は、ハードディスク、取り外し可能磁気ディスク948および取り外し可能光ディスク952を採用しているが、この分野の精通者ならば当然に理解されるように、コンピュータがアクセスできるデータを、他のタイプのコンピュータ読取可能媒体に保管しておくことも可能である。そのようなものとしては、磁気カセット、フラッシュメモリカード、デジタルビデオディスク、ランダムアクセスメモリ (RAM)、リードオンリメモリ (ROM)などがあり、これらも例示の動作環境で使用することができる。

【0156】いくつかのプログラムモジュールを、ハードディスク、磁気ディスク948、光ディスク952、ROM 938またはRAM 940に格納しておくことも可能であり、そのようなものの例としては、オペレーティングシステム958、1または2以上のアプリケーションプログラム960、他のプログラムモジュール962、およびプログラムデータ964があるが、これらに限定されない。

【0157】ユーザは、キーボード966やポインティングデバイス968（「マウス」など）などの入力デバイスを通してコマンドおよび情報をコンピュータ930に入力することができる。その他の入力デバイス（図示せず）としては、マイクロホン、ジョイスティック、ゲームパッド、サテライトディッシュ、シリアルポート、スキャナなどがある。上記および他の入力デバイスは、バス936に結合されたユーザ入力インターフェース970を通して処理ユニット932に接続されているが、パラレルポート、ゲームポート、ユニバーサルシリアルバス（universal serial bus – USB）などの、他のインターフェースおよびバス構造で接続することも可能である。

【0158】モニタ972や他のタイプのディスプレイデバイスも、ビデオアダプタ974などのインターフェースを通してバス936に接続されている。モニタのほかに、パーソナルコンピュータは、スピーカやブインタなどの他の周辺出力デバイス（図せず）を装備しているのが代表的であり、これらは出力周辺インターフェース975を通して接続可能になっている。

【0159】コンピュータ930は、リモートコンピュータ982などの、1または2以上のリモートコンピュータとの論理的コネクションを使用したネットワーキング環境で動作させることができる。リモートコンピュータ982は、コンピュータ930に関連して上述したエレメントと特徴の多くまたはすべてを具備することができる。

【0160】図9に示す論理的コネクションとは、ローカルエリアネットワーク（local area network LAN）977と一般的な広域ネットワーク（wide area network WAN）979である。このようなネットワーキング環境は、オフィス、企業内コンピュータネットワーク、インターネット、およびインターネット（the Internet）で普及されている。

【0161】LANネットワーキング環境で使用されるときは、コンピュータ930はLAN977のネットワークインターフェースまたはアダプタ986に接続されている。WANネットワーキング環境で使用されるときは、コンピュータは、WAN979上のコミュニケーションを確立するためのモデム978または他の手段を装備しているのが代表的である。モデム978は内蔵され

ているものと、外付けのものとがあるが、どちらも、ユーザ入力インターフェース970または他の該当メカニズムを通してシステムバス936に接続可能になっている。

【0162】図9に示したのは、インターネットを通したWANの構築例である。インターネット上では、コンピュータ930は、インターネット980上のコミュニケーションを確立するためのモデム978または他の手段を装備しているのが代表的である。モデム978は内蔵されているものと、外付けのものとがあるが、どちらも、インターフェース970を通してバス936に接続されている。

【0163】ネットワーキング環境では、パーソナルコンピュータ930に関連して示されているプログラムモジュールまたはその一部は、リモートのメモリストレージデバイスに格納しておくことができる。一例として、図9に示されているリモートアプリケーションプログラムは、リモートコンピュータ982のメモリデバイスに置かれているが、これに限定されない。理解されるように、図示し、上述してきたネットワークコネクションは例示であり、コンピュータ間の通信リンクを確立する他の手段を使用することも可能である。

【0164】例示の動作環境

図9は、例示のバンド幅メータを実現するのに適している動作環境920の例を示す。具体的には、例示のバンド幅メータは、図9では、任意のプログラム960–962またはオペレーティングシステム958によって実現されている。

【0165】この動作環境は、適している動作環境の一例にすぎず、ここで説明されているバンド幅メータが使用される範囲または機能する範囲が限定されることを意味するものではない。バンド幅メータで使用するのに適している、他の周知のコンピューティングシステム、環境、および／または構成としては、パーソナルコンピュータ、サーバコンピュータ、ハンドヘルドまたはラップトップデバイス、マルチプロセッサシステム、マイクロプロセッサベースシステム、プログラマブルコンシュー マエレクトロニクス、ネットワークPC、ミニコンピュータ、メインフレームコンピュータ、上記システムまたはデバイスを含んでいる分散コンピューティング環境などがあるが、これらに限定されない。

【0166】コンピュータ実行可能命令

例示のバンド幅メータの実装は、プログラムモジュールのように、1または2以上のコンピュータまたは他のデバイスによって実行されるコンピュータ実行可能命令を広い意味でとらえて説明することができる。一般的に、プログラムモジュールとしては、特定のタスクを実行する、または特定の抽象データ型を実装しているルーチン、プログラム、オブジェクト、コンポーネント、データ構造などがある。代表例として、プログラムモジュー

ルの機能は、種々の実施形態において必要に応じて結合し、あるいは分散化することができる。

【0167】コンピュータ読取可能媒体

例示のバンド幅メータの実装は、ある種のコンピュータ読取可能媒体に保管しておくことも、媒体間で受け渡しすることも可能である。コンピュータ読取可能媒体は、コンピュータによってアクセス可能であれば、どのような利用可能な媒体にもすることができる。例を挙げる
と、コンピュータ読取可能媒体としては、コンピュータ記憶媒体と通信媒体があるが、これに限定されない。

【0168】コンピュータ読取可能媒体には、コンピュータ読取可能命令、データ構造、プログラムモジュール、他のデータなどの情報を格納するためにいずれかの方法またはテクノロジで実現された揮発性と不揮発性、取り外し可能と取り外し不能媒体がある。コンピュータ記憶媒体としては、RAM、ROM、EEPROM、フラッシュメモリや他のメモリテクノロジ、CD-ROM、デジタルバーサタイルディスク (digital versatile disk DVD) や他の光ストレージ、磁気カセット、磁気テープ、磁気ディスクストレージや他の磁気ストレージデバイス、または必要とする情報を保管しておくために使用でき、コンピュータによってアクセス可能な他の媒体があるが、これらに限定されない。

【0169】通信媒体は、コンピュータ読取可能命令、データ構造、プログラムモジュール、あるいは他のデータを、搬送波などの変調データ信号または他のトランスポートメカニズムの形で具現化しているのが代表的であり、その中には、情報配達媒体が含まれている。ここで「変調データ信号 (modulated data signal)」とは、その特性1つまたは2つ以上が、信号の中の情報を符号化 (エンコード) するような形でセットまたは変更されている信号のことである。例を挙げると、通信媒体には、ワイヤドネットワークや直接ワイヤドコネクションなどのワイヤド媒体、音響、RF、赤外線、その他のワイヤレス媒体のようなワイヤレス媒体があるが、これらに限定されない。上記を任意に組み合わせたものも、コンピュータ読取可能媒体の範囲に含まれている。

【0170】結論

TCPネットワーク環境におけるバンド幅の高速動的測定を、構造上の特徴および/または方法で表したステップに特有の表現で説明してきたが、当然に理解されるように、請求項に明確化されている、TCPネットワーク環境におけるバンド幅の高速動的測定は、必ずしも上述した具体的特徴またはステップに限定されるものではない。むしろ、具体的特徴とステップは、請求項に記載の、TCPネットワーク環境におけるバンド幅の高速動的測定を実現する好適形態として開示されている。

【0171】

【発明の効果】以上説明したように、本発明によれば、Nagle アルゴリズムの適用を禁止すること、「パッシュ」パケットをパケットペアの直後に送信することによってパケットのバッファリングを最小限にすること、およびダミーパケットで初期化 (プライミング) することによって Slow Start アルゴリズムを回避する。

【図面の簡単な説明】

【図1】代表的な公衆ネットワーキング環境（インターネットなど）およびサーバからクライアントに送信されるデータパケットのルーティングと遅延を示す図である。

【図2】Webページのカッタウェイ部分を示す図であり、このカッタウェイには、バンド幅を選択するためのメカニズムをユーザーに提供するユーザインターフェースが示されている。また、そこには、バンド幅を判断するための従来の手法が示されている。

【図3】時間ドメインでグラフ化されたパケットペア（サーバからクライアントに送信される）を示す図である。

【図4】代表的な公衆ネットワーキング環境（インターネットなど）を示す図であり、そこには、バックツーバックで送信されるペアのパケットが示されている。

【図5】例示のバンド幅測定を方法で実現した例を示すフローチャートである。

【図6】図6は、例示のバンド幅測定の異なる側面を方法で実現した具体例の詳細を示すフローチャートである。

【図7】例示のバンド幅測定の異なる側面を方法で実現した具体例の詳細を示すフローチャートである。

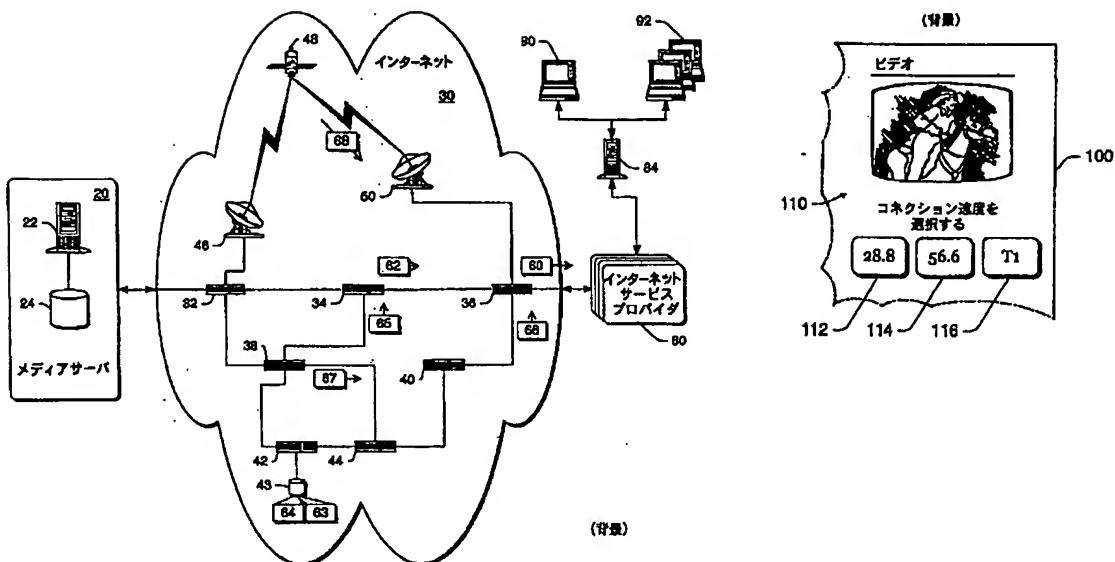
【図8】例示のバンド幅測定の異なる側面を方法で実現した具体例の詳細を示すフローチャートである。

【図9】例示のバンド幅測定の実現を可能にするコンピューティング動作環境の例を示す図である。

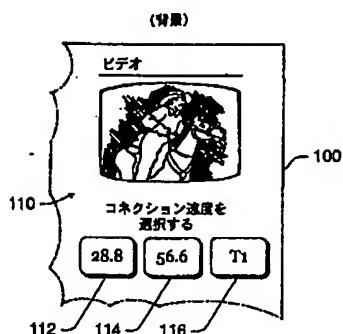
【符号の説明】

- 220 メディアサーバ
- 222 サーバコンピュータ
- 224 データベース
- 230 インターネット
- 232-244 ルータ
- 246、250 サテライトディッシュ
- 248 サテライト
- 260、262 パケット
- 280 インターネットサービスプロバイダ (ISP)
- 282 リンク
- 284 プロキシサーバ
- 290、292 クライアント

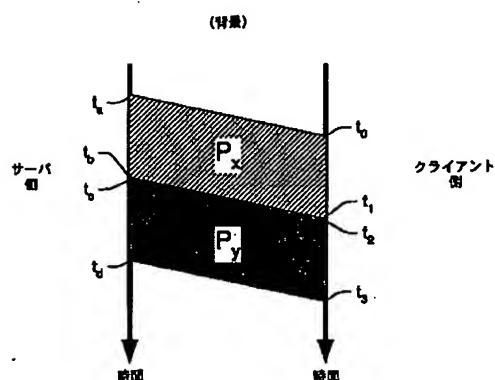
【図1】



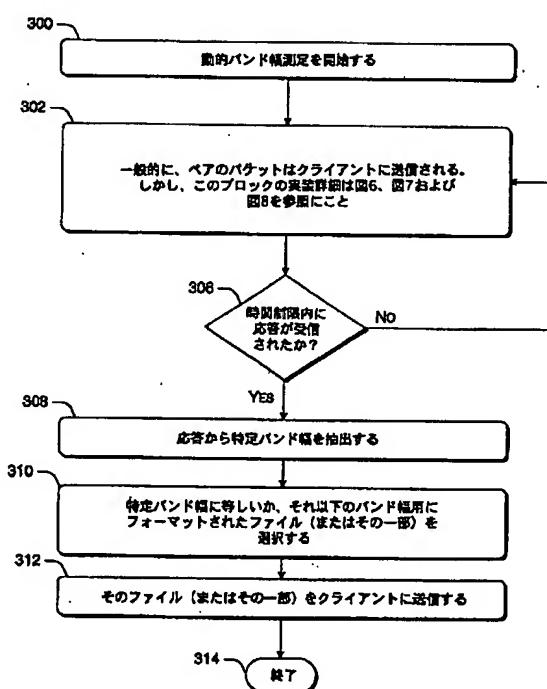
【図2】



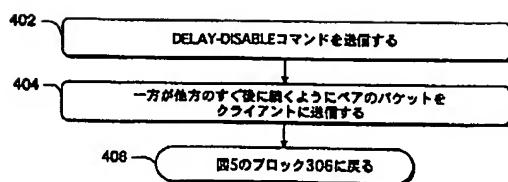
【図3】



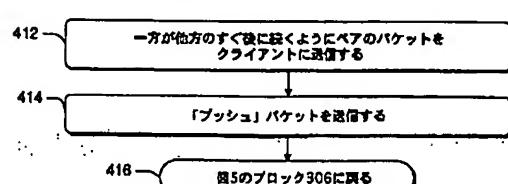
【図5】



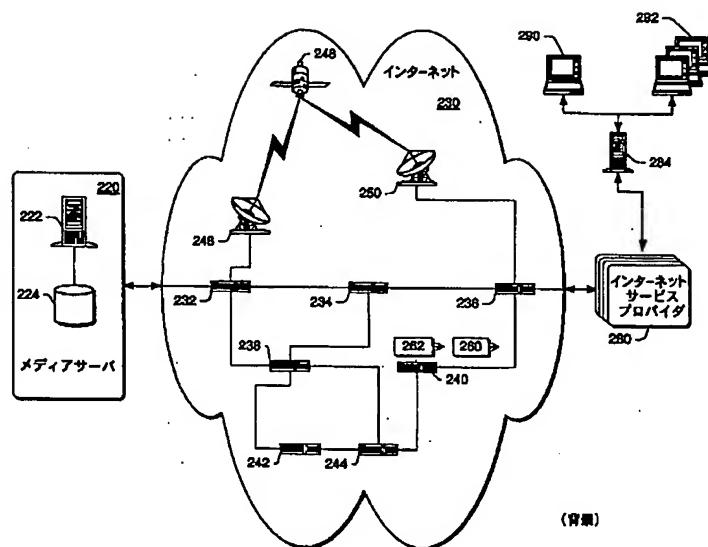
【図6】



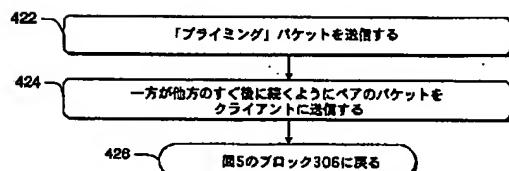
【図7】



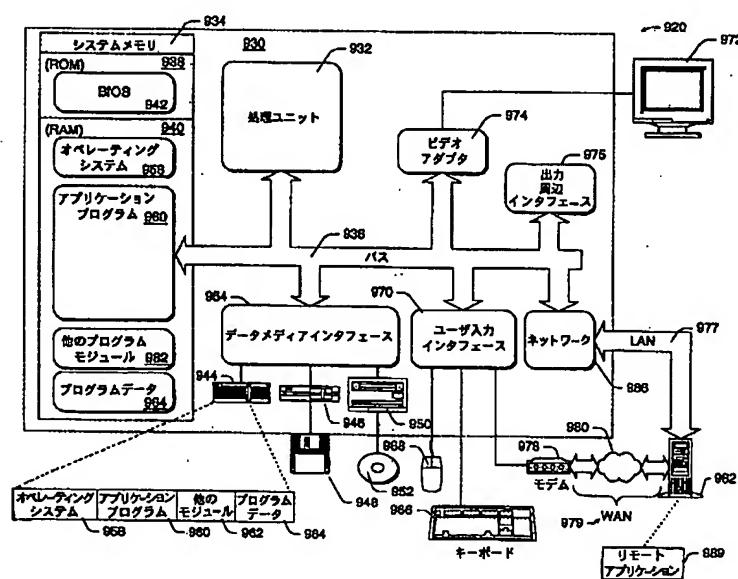
【図4】



【図8】



【図9】



(21) 月2002-94567 (P2002-945U5

フロントページの続き

(72)発明者 ダビド デル バル
スペイン 28034 マドリード バジェ
デ ラシアナ 13

(72)発明者 アンダース イー. クレメッツ
アメリカ合衆国 98122 ワシントン州
シアトル 28 アベニュー 523
Fターム(参考) 5K030 GA02 HA08 LC01 LC14 MB06